Term Project: Different Approaches to Image Colorization

Yuxuan Bao, Xiyuan Chen, Chenkai Sun, Kai Xiong, Xinghao Yu

Abstract

In this project, we investigate the performance of two different approaches on the colorization of manga images. Other than the manga dataset we collected ourselves, we also tested the performance of our first approach on the Labelled Faces in the Wild dataset (LFW). By comparing the results, we can see how different approaches may be more suitable for different kinds of dataset, and which approach performs better on colorizing manga images.

1. Introduction

In this project, we aim to investigate the performance of two different approaches on solving the colorization problem for Manga images. Originated from graphic novels in the late 19th century Japan, Manga has been continuing to gain more and more popularity all around the world in recent decades. Although the traditional Japanese mangas are still mostly drawn in black-and-white, there has been growing attention on the job of colorizing Manga images mainly for two reasons. Firstly, colored comics are more preferred among the western audience from the United States and other countries. Secondly, the growing number of animes based on mangas also require lots of manga image colorization. Therefore, a good approach on generating such colorizations can greatly facilitate the work of animators.

1.1. VAE+MDN Colorization

The two methods we investigate are Variational Autoencoder (VAE) + Mixture Density Network (MDN) Colorization and cGAN-based Colorization. We reckon that these two computer vision methods are suitable and promising in the scope of Manga image colorization.

As a generative model, VAE has the ability to generate lower-dimensional feature embeddings. And when combined with MDN, it has significant versatility colorization problems because provided a single Manga image, each sampling from the MDN network can offer a distinct plausible colorization outputs. It provides variation in colorization styles for the animator to choose from.

1.2. cGAN based Colorization

Like the original GAN, it has a generator model that generates new plausible examples from gray images and a discriminator model that is responsible for identifying the authenticity of a color image.

With the generator and discriminator structure, the cGan model promotes nonlinearity and also eliminates the issue of overfitting, which is significant for manga image colorization, since the images in the manga collection often involve a large number of different characters and the same character would often appear in images in different backgrounds.

In cGAN, the discriminator and generator are conditioned on the gray image and thus we can perform the conditioning by feeding the gray image into both the discriminator and generator. The advantage of cGAN over unconditional GAN is that it has control over the types of images that are generated.

2. Contributions

2.1. VAE + MDN Colorization

2.1.1 Idea/Algorithm

We implemented the combined model of Variational Autoencoder and Mixture Density Network for colorization problem. The idea of using this method to generate multiple instances/styles of colorization from the same input image was based on the paper "Learning Diverse Image Colorization" from CVPR 2017 [1].

2.1.2 Code

Our implementation details, including the code for both VAE and MDN parts are our original work. The paper's implementation was in Tensorflow and it fetched a pretrained model (Zhang et al. colorization network) for MDN features. Our implementation does not make use of any pretrained models. We implemented in Pytorch and developed our own layers for Variational Autoencoder and Mixture Density Network.

2.1.3 Dataset

Before testing its performance on manga images, we firstly trained and tested our model on the same Labelled Faces in the Wild dataset (LFW) as theirs in the paper. This is because our own manga dataset contains fewer images, but the LFW dataset contains a total number of more than 13,000 images of faces collected from the web. And these images are aligned with deep funneling, which means that all images would have some shared structures.

Then we also tried the model on our Manga dataset (details are further explained in the Dataset part) for comparison with the performance of the cGAN-based approach.

2.2. cGAN-based Colorization

2.2.1 Algorithm

We implemented the cGAN based colorization of manga based on the idea from [2], which is modifying the original vanilla cGAN to generate manga images by conditioning on grayscale image as the class.

2.2.2 Code

We implemented the pytorch version code of the algorithm, which is not found anywhere online.

2.2.3 Dataset

We crawled the manga images from a manga website, breaking them up by story plots, and used them to train the model. We generated the colorized images, and compare it with the authors actual drawing in the experiment section.

3. Dataset

The main dataset we test on is the manga named "Kimetsu no Yaiba", which talks about the warriors who eliminate ghosts. The images were crawled from a Chinese Manga website "https://www.mkzhan.com/". We implemented the crawler in Python, using *Beautifulsoup* Library. We first connect to the website using *requests*, extracting the website to a *Beautifulsoup* object. After finding the html tags for manga title links, we request a new page for each link and download the images from the page, by finding the corresponding classes and ids. For preprocessing, we resize the image to 512 by 512. We kept only the colorful images downloaded, since we can convert the colorful image to their black and white correspondents using cv2.

The reason we choose this dataset was that the characters are especially colorful, and is therefore intuitively hard for the model to colorize; the effectiveness of the model will thus be more obvious. The typical coloring is shown in 5.2

Figure 1. Kimetsu no Yaiba colored by the author

4. Method

4.1. VAE+MDN

The first approach our group takes is developing a combined model of Variational Autoencoder (VAE) + Mixture Density Network (MDN) for colorization. Two group members have worked on this method. Instead of generating one single colorization out of every grayscale image input, we make use of the sampling from Mixture Density Network (MDN) to output multiple different plausible colorings for a single input at the same time.

The training process is taken in 2 steps[1]:

Firstly, we train a variational autoencoder on colored images, and generates their feature embeddings in low dimensions.

Secondly, we use a Mixture Density Network to model the conditional distribution of lower-dimensional embeddings over original grayscale images as a mixture of Gaussian distributions. Thus, sampling from the MDN network can generate different feature embeddings out of the same grayscale image. Since theoretically every distribution can be expressed as a mixture of Gaussians, this method is capable of representing arbitrary distributions.

The testing procedure is sampling from the conditional model and then going through the decoder to general several plausible colorization outputs from one grayscale input. These outputs are then evaluated by human observation as well as comparison with outputs from other methods.

The following part gives a detailed description of each part of our model.

Our data for this method needs to have a large number of

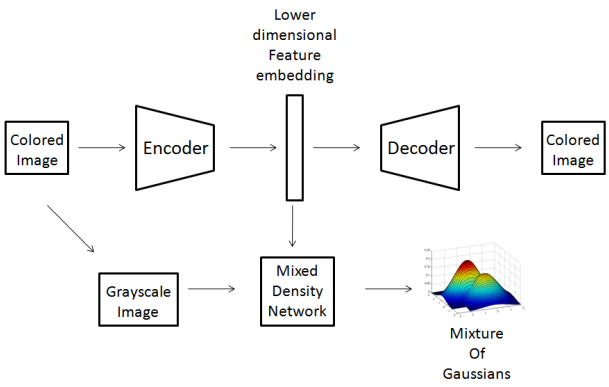


Figure 2. Training Procedure for VAE + MDN

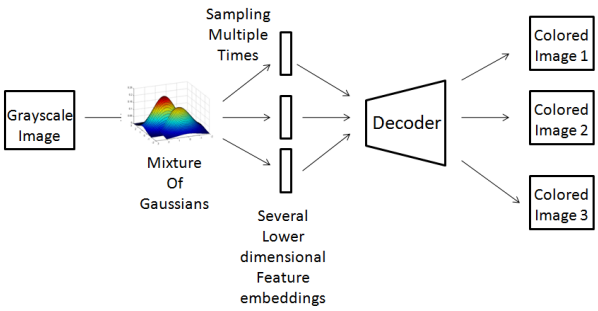


Figure 3. Testing Procedure for VAE + MDN

colored images. The colored versions of them are used to train the variational autoencoder, and the grayscale versions of them are used in training the mixture density network. Before testing our model's performance on manga images, we firstly tried it on face images. We've collected 13000 images from LFW (Labeled Faces in the Wild) dataset and have also converted them to grayscale in preparation for the MDN.

4.1.1 Variational Autoencoder

We've implemented the following structure for the VAE network:

Encoder -

The encoder takes in a color field of 64*64*3 and outputs a 32-dimensional feature embedding. In the original paper, it takes in color field of 64*64*2, which contains the a-channel and b-channel of LAB color space. And in our implementation, we chose a more usual version of RGB color space.

A ReLU activation function is followed after each layer, and the output is converted to a 32-dimensional vector.

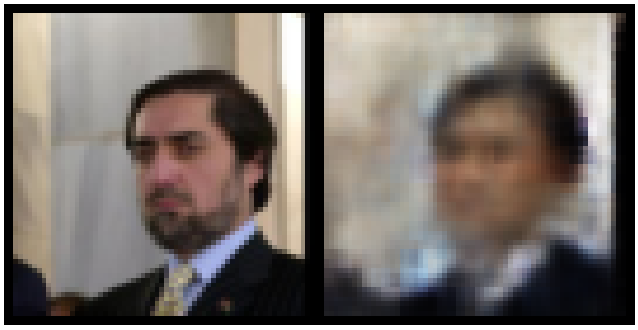
Decoder -

The decoder takes in a 32-dimensional feature embedding and outputs a color field of 64*64*3.

A ReLU activation function is followed after each layer.

We trained on 20000 images with a batch size of 32, using MSELoss and Adam optimizer. We trained for 100 epoches and the followings present some newly generated images by our VAE model with the corresponding original images.

The following image provides an example of the result from our VAE. The shape of the face is recognizable, but the blurring effect shows that the image isn't restored well enough.



4.1.2 Mixture Density Network

Our Mixture Density Network models takes inputs from the trained VAE in the upper part. It takes in the grayscale version of the original colorful images as x, and takes in their corresponding lower-dimensional embeddings generated by the encoder as y. Then, it trains the $3 \times M$ parameters $(W_i)_{i=1}^M$, $(\mu_i)_{i=1}^M$, $(\sigma_i)_{i=1}^M$ of the conditional distribution

$$\mathbb{P}(y|x) = \sum_{i=1}^{M} W_i \times Gaussian(\mu_i, \sigma_i^2)$$

It approximates the underlying distribution by the mixture of M Gaussian distributions. The parameters are trained in a neural network consisting of two parts: one returns the weight parameters $(W_i)_{i=1}^M$, and the other returns the mean and variance parameters $(\mu_i)_{i=1}^M$ and $(\sigma_i)_{i=1}^M$. Each part consists two linear modules and a nonlinear activation function in between.

The input grayscale image x is flattened out to have dimension 4096, and the output feature embedding y has dimension 32. Each input and output dimension is trained seperately to maximize the log likelihood, and the result turns to be fairly good if the number of Gaussian distributions M is set to be high enough. Otherwise, if M is too small, the mixture of Gaussians will not be able to model the complex conditional distribution.

To visualize the model performance, we take an arbitrary grayscale image as x and sample multiple feature embeddings y from the trained Mixture Density Network. We then feed these embeddings into the decoder of the VAE part to generate plausible colorizations of the original grayscale image.

As an example, we trained the MDN on 100 images for 2000 epoches. The following graph presents some grayscale images and the corresponding colorization of them by our VAE + MDN model.

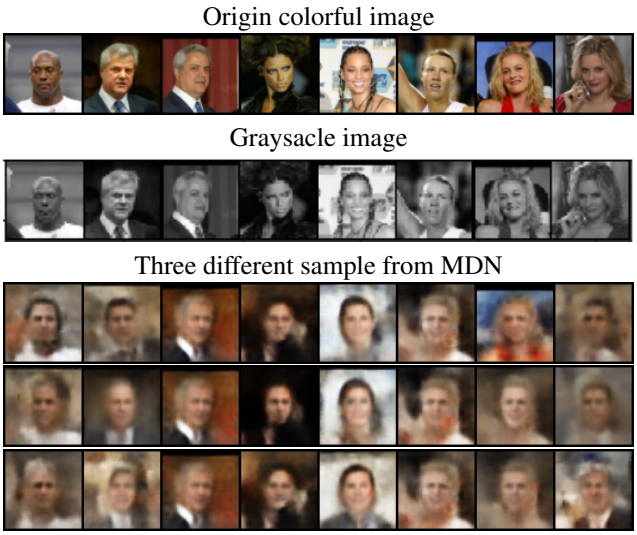


Figure 4. MDN samples

4.2. cGAN

The second approach our group have for colorizing menga images is using conditional Generative Adversarial Networks (cGAN)[2]. Three group members have worked on this method. The cGAN model consists of two neural networks, a generator followed by a discriminator. For each RGB image in the dataset, we create a corresponding gray scale image. Our GAN model is conditioned on the gray scale image. In the training procedure, the discriminator takes two pairs of (gray image, color image), one of the color image being the real one which the other generated by the generator and predict whether the color image is a real or generated one. The process can be seen as in Figure 5.

4.2.1 Generator

For the generator, it takes a gray scale image of 512*512*1 as input and produces a 512*512*3 RGB image. It aims to generate vivid color image which is indistinguishable from the real RGB image by the discriminator. For the loss of the generator, we use Binary Cross-Entropy Loss to evaluate its performance in deceiving the discriminator, and Cosine

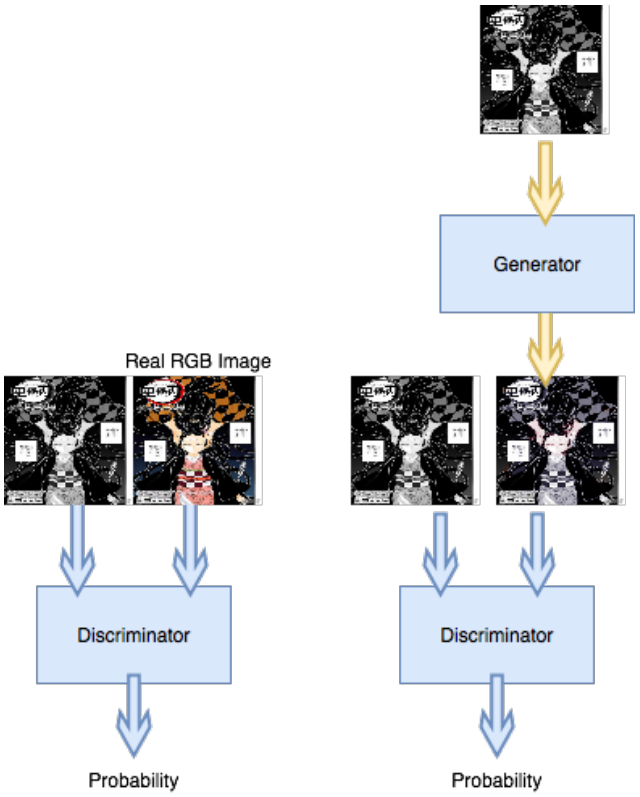


Figure 5. Training Procedure for cGAN

Similarity, Mean Square Error with Mean Absolute Error to evaluate its performance in creating a rgb image similar to the real one.

$$l_1 = 5 * \left(-\frac{1}{N}\sum_{i=1}^{N}\log(p(y_i))\right)$$

$$l_2 = 100 * \left(\frac{1}{N}\sum_{i=1}^{N}\left\|Im - \widetilde{Im}\right\|_2^2 + (1 + \cos\_sim(Im\_batch, \widetilde{Im}\_batch)) * \frac{1}{N}\sum_{i=1}^{N}||Im - \widetilde{Im}||\right)$$

$$l = l_1 + l_2$$

For all N RGB images it creates, $p(y_i)$ is the probability of the RGB imge being real predicted by the discriminator. Im is the real RGB image and $\tilde{\text{I}}$m is the RGB image created by the generator. Im_batch and $\tilde{\text{I}}$m_batch are the flattened vectors of real RGB images in a batch and RGB images created by the generator in the same batch respectively.

We've implemented the following structure for the Generator network:

4.2.2 Discriminator

For the discriminator, for each pair of (512*512*1 gray image, 512*512*3 color image) in the input (color image can be either the real RGB image or the one generated by the

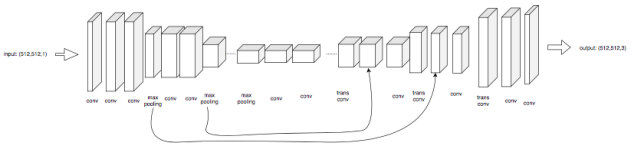


Figure 6. Generator Network Architecture

generator), it outputs a probability of whether the color image is real for the given gray scale image. The discriminator aims to distinguish between the real RGB image and the RGB image produced by the generator. We use Binary Cross-Entropy Loss for the discriminator.

$$-\frac{1}{N}\sum_{i=1}^{N} y_i * \log\big(p(y_i)\big) + (1-y_i) * \log\big(1-p(y_i)\big)$$

For all N input pairs of (gray img, RGB img), y_i is the label (1 for the input pair(gray img, real RGB), 0 for the input pair(gray img, RGB created by generator)), $p(y_i)$ is the predicted probability of the RGB imge being real.

We've implemented the following structure for the Discriminator network:

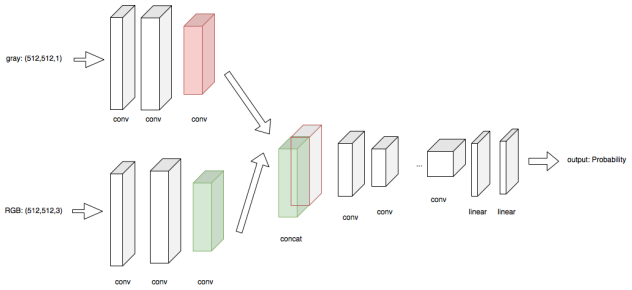


Figure 7. Discriminator Network Architecture

5. Experiments

5.1. VAE + MDN Colorization

The model produces multiple different colorings of every single grayscale image. Each of these colored images can be compared with the original (real) image and have their L2 distance calculated. And if the minimal L2 distance of them all is less than a certain threshold, we can regard this colorization as being successful. Thus, the successful rate can be taken as metric for how good the model performs on colorization. Specifically, our model runs fast and well on LFW dataset whose images are aligned well with deep funneling. The colorization of the facial parts of our result are generated to be better than the background. The reason of it could be that all of the faces share a same structure since those images are aligned well. The resulting colorized images look blurred, and it is due to the blurring effect after the VAE process, as already shown in the previous part.

In order to compare with our second approach, we try to generate the colorized manga image using our training dataset which has about 200 images on it, the result looks much worse than using the LFW dataset. One reason could be the lack of data. Our second cGAN-based approach doesn't need lots of images for training process, but our first VAE + MDN approach does need lots of images to feed into the model. Otherwise, the distribution learned by MDN may fail when being generalized to new data. Also, the patterns of the manga images are not well aligned and structured as the images from LFW dataset. For instance, human body parts may appear anywhere in the picture. The grids and lines would also make the training process for autoencoder a lot harder, and the conditional distribution learned by MDN not very generalizable. For comparing the result from the cGAN, we use the same manga to generate the colorful image. However, due to lacking of training image, it performs far way worse than cGAN of which loss is nearly 5,000.

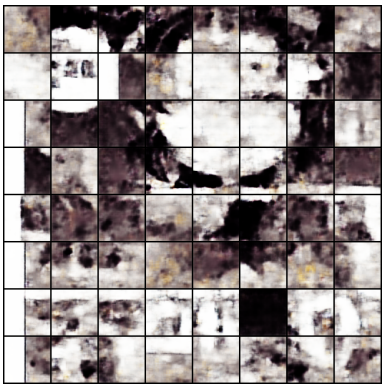


Figure 8. Image Colorized by VAE+MDN

5.2. cGAN-based Colorization

Given a monochrome image of a certain manga character we trained, this model will produce a colored version of it. And the evaluation for this approach is similar to the first approach, where we can compare it with the colorized reference and calculate their Mean Square Error. We will also try it on multiple characters to test its robustness.

During the experiment, we printed output images as training goes on in order to keep track of the model's performance. In the first stage, the model is trying to capture basic shape of the figure in the gray image, but the color looks strange and also blurred, and it kept bouncing between different colors. In the second stage, the model started to produce gray-like images, which has the exact shape of original

image and it became very clear. But it still did not seem to learn the colorization, and the model was stuck there for a long time. The model finally started to show the color after about 200 epochs, it learned to colorize the face of the figure pretty well, but the overall color was relatively dark compared to the colorized version by author. The result is shown below.

Figure 9. Image Colorized by Generator (left) and the Author (right)

For quantitative result, the Mean Square Error between the two images is 482, which is much better than the VAE+MDN.

6. Conclusion

In conclusion, we investigated the performance of two different approaches, customized Variational Autoencoder (VAE) + Mixture Density Network (MDN) Colorization and cGAN-based Colorization to solve the colorization problem for Manga images. We collected dataset from both crawling and existing dataset. In the experiment, we compared the two models in experiment section, with both qualitative and quantitaive measure into the metric, and we eventually found the does a better job at colorizing manga images. In the future, we will try to modify the cGan approach so that it will turn the manga into different style (e.g. more vibrant, more gloomy, etc.).

References

[1] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, and David A. Forsyth. Learning diverse image colorization. *CoRR*, abs/1612.01958, 2016.

[2] Paulina Hensman and Kiyoharu Aizawa. cgan-based manga colorization using a single training image. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 3, pages 72–77. IEEE, 2017.